

REVISIONS TO CLAIMS

A/

1 1. (original) A method of determining the flow of a data object in a software architecture using
2 queues to organize the transfer of data from one processing object to another, comprising the
3 steps of:

4 storing queue identifiers in a path object;
5 receiving and processing a data object in a first of said processing objects;
6 identifying a queue corresponding to a second of said processing objects responsively to
7 an indicator corresponding to said data object;
8 placing said data object in a queue identified in said step of identifying.

2. (original) A method as in claim 1, wherein said step of identifying includes determining a
result of said step processing.

3.(original) A method as in claim 2, wherein said step of identifying includes determining a
result of said step processing and said result corresponding to said queue.

1 4. (original) A method for determining the flow of data in a software architecture in which
2 queues are used to organize the transfer of data from one process to another process, comprising
3 the steps of:

4 performing a process on a data part of a first data object, by a first processing object;
5 identifying a first queue to which said first data object is to be transferred from a
6 indicator part of said first data object;
7 modifying said indicator part of said first data object to produce a second data object;

REVISIONS TO CLAIMS

8 performing said process on said second data object;
A/ 9 identifying a second queue to which said second data object is to be transferred.

1 5. (original) A method as in claim 4, further comprising determining a result of said step of
2 performing, said step of identifying including identifying said second queue responsively to said
3 step of determining.

1 6. (original) A pipeline software architecture in which data objects are transferred from a first
2 processing object to a selected one of second and third processing objects by queuing the data
3 objects in a queue of said selected one, comprising:
4 a definition of a path object corresponding to each of said data objects;
5 at least one of said path objects containing an indicator of at least one of said second and
6 third processing object;
7 said first processing object defining a process a result of which is to insure that a first
8 data object processed by said first processing object is placed in a queue of said at
9 least one of said second and third processing objects responsively to one of said
10 path objects corresponding to said first data object.

1 7. (original) An architecture as in claim 6, wherein said process includes the generation of an
2 indication of a result of a subprocess of said first processing object and said first data object
3 processed by said first processing object is placed in said queue of said at least one of said
4 second and third processing objects responsively to one of said path objects corresponding to
5 said first data object and responsively to said indication.

REVISIONS TO CLAIMS

1 8. (new) In an object oriented programming environment, a method comprising executing the

A2 2 following operations in at least one data processing device:

3 • maintaining a data object in a first queue according to a queue indicator associated with the
4 data object;

5 • responsive to the queue indicator, processing the data object with a first processing object;
6 and

7 • responsive to the processing, changing the queue indicator to indicate a second queue
8 destined for a second processing object;

9 whereby the data object determines its own destiny.

9. (new) The method of claim 8, wherein the queue indicator is stored in a path object associated with the data object and the processing comprises querying the path object.

10. (new) The method of claim 9, wherein the path object includes a table of queue indicators.

1 11. (new) The method of claim 8, wherein

2 • the processing comprises determining a normal or faulty outcome state of the data object; and

3 • the changing is dependent on said normal or faulty outcome state.